Department of Mathematics
University of California
Berkeley, California 94720

Simple approach to $O(p^3)$ scaling in the fast multipole method

Kevin K. Lin
kkylin@math.berkeley.edu

Abstract

This article presents a simple modification to Greengard's fast multipole method that yields essentially $O(Np^3)$ running time for pth-order N-particle calculations. This method is compared to other $O(Np^3)$ methods that have been proposed.

1 Introduction

Fast multipole methods enable accurate and efficient calculations of Coulombic energies and forces in particle systems: The dominant part of the algorithm scales linearly with the number of particles, N, rather than N^2 as in direct summation.[1] It has important implications for a wide range of applications, including molecular simulations, N-body simulations in celestial mechanics, and particle methods in fluid mechanics,[2] and has received much attention in the literature [2, 4, 5, 10, 12]. However, in Greengard's original formulation [5], a pth-order method requires $O(Np^4)$ operations in the multipole calculation stage, limiting the method's usefulness in problems requiring intermediate to high accuracy.

Fast multipole methods rely extensively on "translation operators" to shift the centers of multipole and local (Taylor) expansions. These operations, if implemented in a straightforward way, require $O(p^4)$ operations on each invocation. In [11], White and Head-Gordon introduced the use of Wigner rotation matrices to reduce the running time of each translation (and hence that of the fast multipole algorithm) from $O(p^4)$ to $O(p^3)$, and in [2], Cheng, Greengard, and Rokhlin present yet another $O(p^3)$ algorithm, based on "plane wave" expansions, that is faster by a constant factor and provides higher accuracy. However, the plane wave expansion of Cheng *et. al.* is rather complicated, while White and Head-Gordon's rotation-based method, though simple conceptually, requires some care to apply correctly and effectively.

[1] As observed by Cheng, Greengard, and Rokhlin in [2], most multipole methods involve a "loading" stage that partitions particles into an octree structure. While the asymptotic worst-case running time of this stage is generally $O(N \log N)$, it is the $O(N)$ multipole computation stage that, in practice, dominates the total running time.

[2] Please see Greengard [5] for more extensive referenes.

In this article, a new $O(p^3)$ algorithm, based on simple recurrence relations for spherical harmonics, due to Pohorille [8], is presented. While it has the same asymptotic running time as that of plane wave or rotation-based methods, this method is, in our opinion, easier to implement and understand, even though it does not offer all the benefits of plane wave expansions.

The rest of the paper goes like this: Section 2 reviews the basic theory underlying multipole expansions, Section 3 summarizes relevant recurrence relations and uses them to develop a new $O(p^3)$ algorithm, and Section 4 compares our new method to other $O(p^3)$ methods and discusses some implementation issues. Readers interested in the details of my implementation will have to wait for [7].

2 Multipole Expansions

This paper follows the notations of White and Head-Gordon [10], which is more transparent and compact than that of Greengard [5]. Relevant results regarding spherical harmonics and multipole expansions are only summarized here; interested readers can consult Arfken [1] or Jackson [6] for details. This paper concentrates on the theory of multipole expansions and their translation operators, and anyone interested in the use of multipole expansions and translation operators in the fast multipole method should see White and Head-Gordon [10], Cheng, Greengard, and Rokhlin [2], or Figueirido *et. al.* [4].

2.1 Basic theory

2.1.1 Multipole expansions

Multipole expansions are a standard tool in the study of electrostatics [6]. They are based on the observation that

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \sum_{l=0}^{\infty} P_l(\cos\gamma)\cdot\frac{r^l}{s^{l+1}}, \tag{1}$$

where the P_l are Legendre polynomials and γ denotes the angle between $\mathbf{r}$ and $\mathbf{s}$. Upon rotating the coordinate system, we get

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \frac{(l-m)!}{(l+m)!}\cdot P_{l,m}(\cos\theta_r)\cdot P_{l,m}(\cos\theta_s)\cdot e^{im(\phi_s-\phi_r)}, \tag{2}$$

where $\mathbf{r}=(r,\theta_r,\phi_r)$ and $\mathbf{s}=(s,\theta_s,\phi_s)$ in spherical coordinates, and $P_{l,m}$ denotes the associated Legendre function of order (l,m) (cf. Arfken [1]). Now, define

$$O_{l,m}(\mathbf{r}) = \frac{r^l}{(l+m)!}\cdot P_{l,m}(\cos\theta_r)\cdot e^{-im\phi_r} \tag{3}$$

and

$$M_{l,m}(\mathbf{s}) = \frac{(l-m)!}{s^{l+1}}\cdot P_{l,m}(\cos\theta_s)\cdot e^{im\phi_s} \tag{4}$$

so that Equation 2 becomes

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} M_{l,m}(\mathbf{s}) \cdot O_{l,m}(\mathbf{r}). \tag{5}$$

Suppose now we are given a cluster of N charges with positions $\{\mathbf{r}_i\}$ and charges $\{q_i\}$, and we wish to compute the electrostatic potential

$$U(\mathbf{r}) = \sum_{i=1}^{N} \frac{q_i}{|\mathbf{r}-\mathbf{r}_i|} \tag{6}$$

generated by the charge cluster at some point $\mathbf{r}$ far away from the center $\mathbf{r}_0$ of the cluster. Define

$$\hat{O}_{l,m}(\mathbf{r}_0) = \sum_{i=1}^{N} q_i \cdot O_{l,m}(\mathbf{r}_i - \mathbf{r}_0) \tag{7}$$

and observe that

$$U(\mathbf{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} M_{l,m}(\mathbf{r}-\mathbf{r}_0) \cdot \hat{O}_{l,m}(\mathbf{r}_0). \tag{8}$$

The coefficients $\hat{O}_{l,m}(\mathbf{r}_0)$ are the *multipole coefficients* for the potential U, and the expansion (8) is the *multipole expansion* of U. In electrostatic problems, the potential contains all the relevant information about the charges, and hence we may also speak of multipole coefficients for the system of charges $\{(q_i, \mathbf{r}_i)\}$.

Note that the series in (8) converges if and only if $|\mathbf{r}-\mathbf{r}_0|$ is greater than $\max\{|\mathbf{r}_i - \mathbf{r}_0|\}$, so multipole expansions of electrostatic potentials are essentially "far-field" expansions: They give series expansions, centered at a given point $\mathbf{r}_0$, for the effects one would observe when standing far away from $\mathbf{r}_0$. The fast multipole method exploits multipole expansions to represent far-field effects and reserves direction summation, a relatively expensive operation, for near-field interactions.

2.1.2 Local expansions

In writing down Equation 8, we used Equation 2 to represent the electrostatic potential generated at $\mathbf{s}$ by a unit charge at $\mathbf{r}$. But we can also use (2) to represent the potential generated at $\mathbf{r}$ by a unit charge at $\mathbf{s}$: That is,

$$U(\mathbf{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \hat{M}_{l,m}(\mathbf{r}_0) \cdot O_{l,m}(\mathbf{r}-\mathbf{r}_0) \tag{9}$$

where we can define $\hat{M}_{l,m}(\mathbf{r}_0)$ by

$$\hat{M}_{l,m}(\mathbf{r}_0) = \sum_{i=1}^{N} q_i \cdot M_{l,m}(\mathbf{r}_i - \mathbf{r}_0). \tag{10}$$

This is dual to the multipole expansion (8), and is the *Taylor expansion* of U about $\mathbf{r}_0$: In contrast to multipole expansion (8), this series converges if and only if $|\mathbf{r}-\mathbf{r}_0|$ is *smaller* than $\min\{|\mathbf{r}_i-\mathbf{r}_0|\}$. Like the multipole expansion, it is again a far-field expansion. Unlike multipole expansions, the charge cluster must be far away from $\mathbf{r}_0$ and the point $\mathbf{r}$ close to $\mathbf{r}_0$, rather than the other way around. In this context, Taylor expansions (9) are often known as a *local expansions* (and their coefficients *local coefficients*) for U about $\mathbf{r}_0$.

In fast multipole methods, multipole expansions (as in (8)) are used as intermediate representations for computing local expansions. The evaluation of the potential U is actually done through local Taylor expansions (9).

2.1.3 Conjugate symmetry

Readers familiar with quantum mechanics or, equivalently, representations of classical Lie groups, will recognize that multipole expansions involve so-called solid harmonics, which are closely related to eigenfunctions of quantum mechanical angular momentum operators [9]. It is well known that the associated Legendre functions $P_{l,m}$ (and hence $O_{l,m}$ and $M_{l,m}$) are nonzero only for $l \geq 0$ and $|m| \leq l$. It is also useful to note (see [1]) that

$$P_{l,-m}(x) = (-1)^m \cdot \frac{(l-m)!}{(l+m)!} \cdot P_{l,m}(x) \tag{11}$$

so that

$$O_{l,-m} = (-1)^m \overline{O_{l,m}} \tag{12}$$

and

$$M_{l,-m} = (-1)^m \overline{M_{l,m}} \tag{13}$$

hold. (Overline denotes complex conjugation.) This symmetry saves storage costs by nearly a factor of two in implementations of multipole methods.

2.2 Translation operators

The fast multipole method relies heavily on the following identities:

1. *Multipole-to-multipole translation:*

$$O_{l,m}(\mathbf{r}+\mathbf{s}) = \sum_{j=0}^{l} \sum_{k=-j}^{j} O_{l-j,m-k}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}) \tag{14}$$

2. *Local-to-local translation:*

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = \sum_{j=l}^{\infty} \sum_{k=-j}^{j} O_{j-l,k-m}(\mathbf{s}) \cdot M_{j,k}(\mathbf{r}), \tag{15}$$

3. *Multipole-to-local translation:*

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = (-1)^l \sum_{j=0}^{\infty} \sum_{k=-j}^{j} M_{j+l,k+m}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}), \tag{16}$$

References [5, 2, 10, 11] contain more details regarding these identities, but readers should keep in mind that the third equation, as it appears in [10, 11], does not have the phase factor $(-1)^l$ and is therefore incorrect.[3]) Also, the first two equations (multipole-to-multipole and local-to-local) can sometimes involve to coefficients $O_{l',m'}$ with $|m'| > l'$: Such coefficients are always set to zero.

These linear operators derive their name from the fact that they can be used to shift the centers of multipole and local expansions.[4] For example, suppose we have, as before, a cluster of particles with charge $\{q_i\}$ and positions $\{\mathbf{r}_i\}$. Let U, as before, denote the electrostatic potential generated by these charges, and suppose $\mathbf{r}_0$ and $\mathbf{r}'_0$ are points near the center of the cluster while $\mathbf{r}$ is a point far away from the cluster. We then have two multipole expansions:

$$\begin{aligned} U(\mathbf{r}) &= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} M_{l,m}(\mathbf{r}-\mathbf{r}_0) \cdot \hat{O}_{l,m}(\mathbf{r}_0) \\ &= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} M_{l,m}(\mathbf{r}-\mathbf{r}'_0) \cdot \hat{O}_{l,m}(\mathbf{r}'_0) \end{aligned}$$

where $\hat{O}_{l,m}$ is defined by (7).

We can then convert between the two expansions using

$$\hat{O}_{l,m}(\mathbf{r}'_0) = \sum_{j=0}^{l} \sum_{k=-j}^{j} O_{l-j,m-k}(\mathbf{r}_0-\mathbf{r}'_0) \cdot \hat{O}_{j,k}(\mathbf{r}_0), \tag{17}$$

since $(\mathbf{r}_i-\mathbf{r}_0)+(\mathbf{r}_0-\mathbf{r}'_0) = (\mathbf{r}_i-\mathbf{r}'_0)$ (see the definition of $\hat{O}_{l,m}$ in (7)). Similarly, we can translate or shift local expansions between different centers, and convert multipole expansions from a nearby center to a local expansion about a distant point.

In multipole calculations, the user typically begins by choosing a value of p for which the approximation gives sufficient accuracy. Then, the infinite series in

[3] There are simple tricks to check for certain types of errors in these translation operators. For example, $M_{j,k}(\mathbf{s})$ contains $s^{-(j+1)}$ and the translated result needs to contain $|\mathbf{r}-\mathbf{s}|^{-(l+1)}$. This tells us that, by essentially dimensional analysis, the translation operator should have elements that scale like r^{j-l}, so that $-(j+1)+(j-l) = -(l+1)$ (rather than $l-j$ or other mistakes one might make). One can similarly check that the exponential phase factors have exponents that "add up." Unfortunately, this kind of analysis does not catch the kind of error where the factor $(-1)^l$ was omitted.

[4] The words "translate" and "shift," in this context, are used interchangeably, and will always mean shifting the center of a multipole or local expansion from one point to another, *or* to convert a multipole expansion to a local one via Equation 16.

(9) is truncated to retain only terms for which $l \leq p$. The translation operators (15) and (16) are similarly truncated in all caclulations, and it is easy to see that straightforward evaluation of (14), (15), and (16) requires $O(p^4)$ operations. Readers interested in careful error analyses should see [2].

3 Faster Translation Operators

3.1 Basic recurrence relations

Associated Legendre functions satisfy a number of useful recurrence relations, among them:

$$P_{l,l}(\cos\theta) = (2l-1)!! \cdot \sin^l \theta \tag{18}$$

$$(l-m) \cdot P_{l,m} = (2l-1) \cdot \cos\theta \cdot P_{l-1,m} - (l+m-1) \cdot P_{l-2,m} \tag{19}$$

$$P_{l,m} = (2l-1) \cdot \sin\theta \cdot P_{l-1,m-1} + P_{l-2,m} \tag{20}$$

$$(l-m)(l-m-1) \cdot P_{l,m} = -(2l-1) \cdot \sin\theta \cdot P_{l-1,m+1} + (l+m) \cdot (l+m-1) \cdot P_{l-2,m} \tag{21}$$

$$\cos\theta \cdot P_{l,m} = (l-m+1) \cdot \sin\theta \cdot P_{l,m-1} + P_{l-1,m} \tag{22}$$

$$\sin\theta \cdot P_{l,m} = -(l-m+2)(l-m+1) \cdot \sin\theta \cdot P_{l,m-2} + 2(m-1) \cdot P_{l-1,m-1} \tag{23}$$

$$\sin\theta \cdot P_{l,m} = -2(l-m+1) \cdot \cos\theta \cdot P_{l,m-1} + (l-m+2)(l-m+1) \cdot \sin\theta \cdot P_{l,m-2} + 2l \cdot P_{l-1,m-1} \tag{24}$$

$$(l-m) \cdot \cos\theta \cdot P_{l,m} = -\sin\theta \cdot P_{l,m+1} + (l+m) \cdot P_{l-1,m} \tag{25}$$

$$2l \cdot P_{l,m} = (l+m) \cdot (l+m-1) \cdot \sin\theta \cdot P_{l-1,m-1} - \sin\theta \cdot P_{l-1,m+1} + 2(l+m) \cdot \cos\theta \cdot P_{l-1,m} \tag{26}$$

Note that all associated Legendre functions are evaluated at $\cos\theta$ in the table above: $P_{l,m} = P_{l,m}(\cos\theta)$. Also, this table contains more recurrences than necessary, and they are certainly not linearly independent, but the reader may find this table useful in working with multipole methods.

Using recurrences (24) and (26) above and the definitions of $O_{l,m}(r,\theta,\phi)$ and $M_{l,m}(r,\theta,\phi)$, we get:

$$\begin{aligned} 2l \cdot O_{l,m} \quad &= \quad r\sin\theta \cdot e^{-i\phi} O_{l-1,m-1} - \\ &\qquad r\sin\theta \cdot e^{i\phi} O_{l-1,m+1} + 2r\cos\theta O_{l-1,m} \end{aligned} \tag{27}$$

and

$$\begin{aligned} M_{l,m} \quad &= \quad -2 \cdot \cot\theta \cdot e^{i\phi} \cdot M_{l,m-1} + \\ &\qquad e^{i2\phi} \cdot M_{l,m-2} + \frac{2l}{r\sin\theta} \cdot e^{i\phi} \cdot M_{l-1,m-1}. \end{aligned} \tag{28}$$

3.2 Translation operators

Armed with (27) and (28), we can now formulate a faster way to perform translations.

3.2.1 Multipole-to-multipole translation

Equation 14 states that

$$O_{l,m}(\mathbf{r}+\mathbf{s}) = \sum_{j=0}^{l} \sum_{k=-j}^{j} O_{l-j,m-k}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}).$$

It is easy to see that there is some redundancy in the calculations, and here is one way to exploit this redundancy: Define the partial sums

$$R^j_{l,m} = \sum_{k=-j}^{j} O_{l-j,m-k}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}) \tag{29}$$

with $0 \leq j \leq l$ and $|m| \leq l$. Then

$$O_{l,m}(\mathbf{r}+\mathbf{s}) = \sum_{j=0}^{l} R^j_{l,m} \tag{30}$$

and, from (27), we have

$$\begin{aligned} R^j_{l,m} \quad &= \quad \frac{1}{2(l-j)} \Big[s\sin\theta_s \cdot e^{-i\phi_s} \cdot R^j_{l-1,m-1} - \\ &\qquad s\sin\theta_s \cdot e^{i\phi_s} \cdot R^j_{l-1,m+1} + 2s\cos\theta_s \cdot R^j_{l-1,m} \Big]. \end{aligned} \tag{31}$$

This recurrence allows us to compute all the partial sums $R^j_{l,m}$, except those with $m = 0$, $m = l$, and $m = l-1$. Thus, we need only compute $R^j_{l,0}$, $R^j_{l,l-1}$ and $R^j_{l,l}$ for $0 \leq j \leq l \leq p$ directly, and can compute all other $R^j_{l,m}$ from the recurrence relation. Since there are $O(p^2)$ initial partial sums to compute and they each cost $O(p)$, and since it takes $O(p^3)$ operations to obtain all the partial sums using the recurrence relation, this way of performing multipole-to-multipole translations requires $O(p^3)$ operations.

3.2.2 Local-to-local translation

We can do the same thing with the local-to-local translation operator:

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = \sum_{j=l}^{\infty} \sum_{k=-j}^{j} O_{j-l,k-m}(\mathbf{s}) \cdot M_{j,k}(\mathbf{r}),$$

Define

$$S_{l,m}^{j} = \sum_{k=-j}^{j} O_{j-l,k-m}(\mathbf{s}) \cdot M_{j,k}(\mathbf{r}) \tag{32}$$

with $0 \leq l \leq j$ and $|m| \leq l$. Then

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = \sum_{j=l}^{\infty} S_{l,m}^{j} \tag{33}$$

and, using (27) again, we get

$$\begin{aligned} S_{l,m}^{j} \quad = \quad & \frac{1}{2(l-j)} \Big[s \sin\theta_s \cdot e^{-i\phi_s} \cdot S_{l+1,m+1}^{j} - \\ & s \sin\theta_s \cdot e^{i\phi_s} \cdot S_{l+1,m-1}^{j} + 2s \cos\theta_s \cdot S_{l+1,m}^{j} \Big]. \end{aligned} \tag{34}$$

We can use this recurrence relation to compute all the partial sums $S_{l,m}^{j}$ except $S_{l,0}^{j}$ and $S_{p,m}^{j}$ $(0 \leq l, m \leq p)$. Again, it is easy to see that this requires $O(p^3)$ operations.

3.2.3 Multipole-to-local translation

Finally, let us consider

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = (-1)^l \sum_{j=0}^{\infty} \sum_{k=-j}^{j} M_{j+l,k+m}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}),$$

Define

$$T_{l,m}^{j} = \sum_{k=-j}^{j} M_{j+l,k+m}(\mathbf{s}) \cdot O_{j,k}(\mathbf{r}) \tag{35}$$

with $0 \leq l$, $0 \leq j$, and $|m| \leq l$. Then

$$M_{l,m}(\mathbf{r}-\mathbf{s}) = (-1)^l \sum_{j=0}^{\infty} T_{l,m}^{j} \tag{36}$$

and, using (28) this time, we get

$$\begin{aligned} T_{l,m}^{j} \quad = \quad & -2 \cot\theta_s \cdot e^{i\phi_s} \cdot T_{l,m-1}^{j} + \\ & e^{i2\phi_s} \cdot T_{l,m-2}^{j} + \frac{2(j+l)}{s \sin\theta_s} \cdot e^{i\phi_s} \cdot T_{l-1,m-1}^{j}. \end{aligned} \tag{37}$$

Again, we can use this recurrence relation to compute all the partial sums $T^j_{l,m}$ except $T^j_{l,0}$ and $T^j_{l,1}$.[5] Again, it is easy to see that this requires $O(p^3)$ operations.

3.2.4 Vertical multipole-to-local translation

In the recurrence relations for $T^j_{l,m}$ above, we see that $\sin\theta_s$ appears in the denominator. This means that, when $\sin\theta_s = 0$ (which corresponds to translating the center of the expansion along the $\hat{\mathbf{z}}$-axis), the recurrence is ill-defined. But, in that case, both the recurrence relation and the translation operator itself simplify: (16) becomes

$$M_{l,m}(\mathbf{r} - \mathbf{s}) = (-1)^l \sum_{j=0}^{\infty} \frac{(j+l)!}{s^{j+l+1}} \cdot O_{j,-m}(\mathbf{r}), \tag{38}$$

which only takes $O(p^3)$ to compute all of the $M_{l,m}$.[6] In the fast multipole method, $\sin\theta_s$ is never a small number: Either it is zero, or it is sufficiently large that the recurrence for $T^j_{l,m}$ can be safely used. Thus, it is easy to rewrite one's code to take this into account and use the simplified translation operator whenever $\sin\theta_s = 0$.

One may find the need to handle this special case rather unsatisfying, and may wish to find recurrence relations which do not involve dividing by $\sin\theta_s$ or $\cos\theta_s$. Indeed, it is possible to derive such a recurrence relation for the $T^j_{l,m}$: For example, it is true that

$$\begin{aligned} T^j_{l,m} \quad = \quad & \frac{s\sin\theta_s}{2(j+l+1)} \cdot e^{-i\phi_s} \cdot T^j_{l+1,m+1} - \\ & \frac{s\sin\theta_s}{2(j+l+1)} \cdot e^{i\phi_s} \cdot T^j_{l+1,m-1} + \frac{s\cos\theta_s}{j+l+1} \cdot T^j_{l+1,m} \end{aligned} \tag{39}$$

holds. But numerical experiments and simple analyses both seem to indicate that this particular recurrence relation is numerically unstable, and I was unable to find a stable recurrence of the desirable form. Furthermore, it is not difficult in practice to write the multipole-to-local translation routines so that it always uses the correct form of the translation operator.

4 Implementation Issues & Future Work

4.1 Comparisons with other methods

As stated in the Introduction, White and Head-Gordon [11] and Cheng *et. al.* [2] have both proposed and implemented $O(p^3)$ versions of the translation

[5] Recall that, because of the conjugate symmetry $O_{l,-m} = (-1)^m \overline{O_{l,m}}$, we need only compute multipole and local coefficients for $0 \leq m \leq l$.

[6] Note that this applies only when $\mathbf{s} = s\hat{\mathbf{z}}$, that is, when we are shifting the center of the expansion *upwards*. It is relatively easy, however, to work out a similar expression when shifting *downwards*.

operator, with extensive timing tests. Our method is closer in accuracy and speed to that of White and Head-Gordon: Cheng *et. al.* use a fundamentally different representation of far-field effects to perform translations.

Compared to the method of White and Head-Gordon, our method should provide the same accuracy and comparable speed, since they are mathematically equivalent and involve only a different order of operations. We feel that this method is simpler conceptually, as it requires only elementary mathematics, but that is, in a way, a matter of taste.

On the other hand, the method of Cheng *et. al.* does provide real improvements over our approach and that of White and Head-Gordon: It is more accurate and hence able to perform accurate translations between boxes[7] that are only separated by one box instead of two.[8] This, in turn, drastically reduces the number of multipole-to-local translations one must perform, which is invoked much more often than its multipole-to-multipole and local-to-local counterparts in typical fast multipole calculations. On the other hand, their method is more complicated, as it requires three representations of far-field effects instead of two, and the cost in software development time may be serious. In addition, parallel implementations are desirable in many applications, and in such cases the optimal method may depend strongly on the architecture and method of parallelization: the method of Cheng *et. al.* may very well be more efficient, but one cannot be absolutely sure until all methods have been implemented and compared. Unfortunately, that comparison is beyond the scope of this project.

4.2 Optimality

The translation operators presented above require $O(p^3)$ operations. For massively parallel applications, however, communications costs often outweigh operation counts. In our case, if information about two charge clusters are stored in the local caches of different processors and a multipole-to-local translation is required between these clusters, we would need to transfer $(p+1)^2$ numbers between their corresponding processors. This is the communications cost, for example, of simply sending the coefficients $\hat{O}_{l,m}$ for $0 \leq m \leq l \leq p$.

One may hope to use recurrence relations to first preprocess and "compress" the information in $\hat{O}_{l,m}$ to reduce communication bandwidth requirements, but as Eric Darve [3] points out, the fast multipole method strives to represent far-field effects using the minimum number of degrees of freedom, and thus it would be difficult to "compress" the information they contain. Furthermore, our translation operators depend critically on the existence of recurrence relations that depend only on one index (in our case, l) instead of both indices l and m. As there is only one such recurrence relation for each index, it is unlikely that

[7] The fast multipole method partitions a system of charges into clusters or "boxes," and these two terms are used interchangeably here.

[8] If you don't know what I'm talking about, see the discussion of "well-separated boxes" in [10].

this would work. However, no rigorous proof is known at the moment.[9]

Note that even if one manages to prove the need for $O(p^2)$ information, it does not preclude the possibility of improving the $O(p^3)$ running time to $O(p^s)$ for $s < 3$. Technically, it also does not preclude the possibility of "lossy compression" of information, but any such scheme would probably be equivalent to using lower-order expansions.

4.3 Numerical stability

One important remaining issue is that of numerical stability: In tests with uniform distributions of Coulomb charges, the recurrence relations appeared to be very stable up to $p \geq 40$. But we have not performed tests with highly non-uniform distributions, nor have we obtained rigorous proofs of stability.

Appendix

This is incomplete, so it sits here:

Arguments utilizing the linear independence of (linear) recurrence relations can prove that "compression" cannot be done via recurrences. However, I think an even stronger result is possible: Compression is impossible via any *smooth* function (linear or nonlinear).

First, some terminology: Let M denote the space of all possible configurations for N charges in the unit cube: This is a $3N$-dimensional manifold. Let V denote the space of functions $f : \mathbf{R}^3 \longrightarrow \mathbf{R}$ with the property that Δf is a linear combination of N Dirac delta functions *and* f vanishes at infinity, and let W denote the real vector space $\mathbf{R}^{(p+1)^2}$.[10] As will be explained below, M formally represents all possible configurations of N charges in the unit cube $[0,1]^3$, V is the space of potentials that they generate, and W is the space of truncated multipole expansions of these potentials.

We can map M into the vector space V by solving Poisson's equation

$$-\frac{1}{4\pi}\Delta f = \sum_{i=1}^{N} q_i \cdot \delta_{\mathbf{r}_i}$$

for a given set of positions $(\mathbf{r}_1, \ldots, \mathbf{r}_N)$, with the condition that f vanishes at infinity. This defines a smooth map ι from M into V. Next, by restricting functions f in V to a region outside the cube $[0,1]^3$, we can use multipole expansions to represent f uniquely. This allows us to define a projection operator π from V into W by truncating multipole expansions, and the composite map $\psi = \pi \circ \iota$ maps M into $\mathbf{R}^{(p+1)^2}$.

The key issue is: What is the rank of the derivative $d\psi$ of ψ? If it has full rank, for example, then whenever $3N > (p+1)^2$, ψ would be a submersion and

[9] But see the appendix.

[10] Δ denotes the Laplacian operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.

therefore cannot factor through a lower-dimensional manifold. This would mean that there are no smooth transformations that can map all multipole coefficients into fewer real variables than $(p+1)^2$ and still retain all the information. In terms of communications cost, $O(p^2)$ would then be the best one can do.

It should not be too difficult to prove some rigorous results regarding the rank of the map ψ, but that has not yet been done. Explicit numerical computation of singular values of $d\psi$ for $N \approx 50$ and $p \approx 7$ indicate that $d\psi$ generally has rank $(p+1)^2 - 1$, which means that whenever $3N + 1 > (p+1)^2$ it would be impossible to compress the information. Results of calculating singular values with $N \approx 256$ and $p \approx 30$ are more ambiguous.

Acknowledgments

I am grateful to Andrew Pohorille and Eric Darve for their help with this project, and to Michael New and Michael Wilson for making their personal libraries available.

References

[1] George B. Arfken. *Mathematical Methods for Physicists.* Academic Press, New York, second edition, 1985.

[2] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics*, 1999.

[3] Eric Darve. Private communication.

[4] Francisco Figueirido, Ronald M. Levy, Ruhong Zhou, and B. J. Berne. Large scale simulation of macromolecules in solution: Combining the periodic fast multipole method with multiple time step integrators. *Journal of Chemical Physics*, 106(23), 1997.

[5] Leslie Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, Massachusetts, 1988.

[6] John David Jackson. *Classical Electrodynamics.* John Wiley & Sons, New York, second edition, 1975.

[7] Kevin K. Lin. Implementation notes on the fast multipole method, 2000. Unpublished (in fact, not yet finished), but will be available some time soon.

[8] Andrew Pohorille. Private communication.

[9] Hermann Weyl. *The Theory of Groups and Quantum Mechanics.* Dover Publications, New York, 1950.

[10] Christopher A. White and Martin Head-Gordon. Derivation and efficient implementation of the fast multipole method. *Journal of Chemical Physics*, 101(8), 1994.

[11] Christopher A. White and Martin Head-Gordon. Rotating around the quartic angular momentum barrier in fast multipole method calculations. *Journal of Chemical Physics*, 105(12), 1996.

[12] Feng Zhao. An $O(N)$ algorithm for three-dimensional N-body simulations. Technical Report 995, MIT Artificial Intelligence Laboratory, 1987.